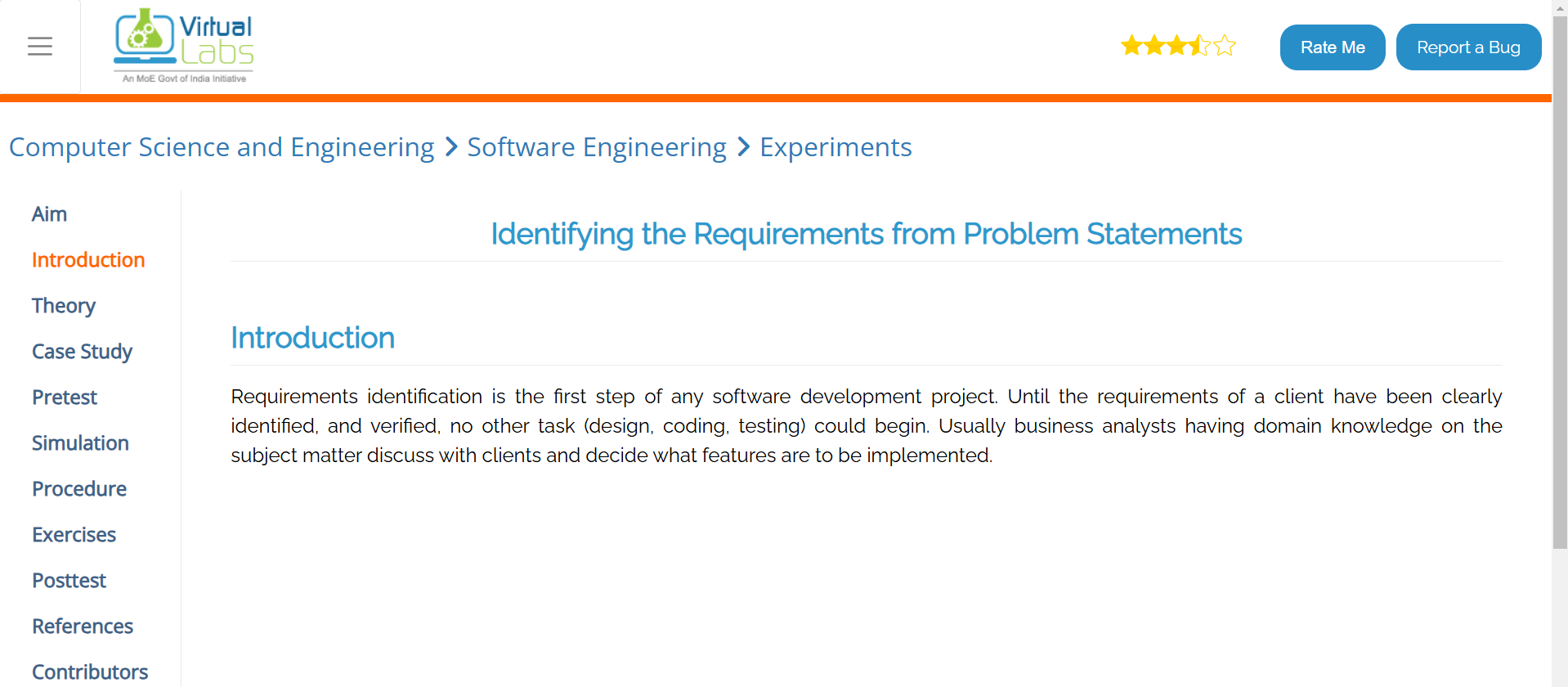
**PRACTICAL NO 1**

**Identifying the functional and non functional requirements for Veterinary hospital DBMS**

**Reference taken from Virtual lab (vlab)**

**Link**[**:-** https://se-iitkgp.vlabs.ac.in/exp/problemstatements/casestudy.html](file:///C:\Users\kalpa\Downloads\-%20https\se-iitkgp.vlabs.ac.in\exp\problemstatements\casestudy.html)

**Software Requirements Specification (SRS) for Veterinary Hospital Database Management System (VDBMS)**

**1. Introduction**

**1.1 Purpose**

The **Veterinary Hospital Database Management System (VDBMS)** is designed to **streamline hospital operations** by automating **pet registration, inpatient and outpatient management, billing, pharmacy tracking, and staff management**. This system will improve **efficiency, accuracy, security, and accessibility** in veterinary hospital operations.

**1.2 Scope**

The VDBMS will manage the following:

* **Pet Registration** – Storing pet details (name, breed, medical history, vaccinations, etc.).
* **Doctor & Nurse Management** – Maintaining staff details and scheduling assignments.
* **Inpatient & Outpatient Care** – Tracking pet visits, diagnoses, and treatments.
* **Billing System** – Automating invoices for services and medication.
* **Pharmacy Management** – Managing medication inventory and prescriptions.
* **User Roles & Authentication** – Ensuring secure access control for doctors, nurses, receptionists, and admins.

**2. Software and Hardware Requirements**

**2.1 Software Requirements**

* **Operating System**: Windows / Linux / macOS
* **Database**: MySQL
* **Backend**: PHP, Node.js, or Python
* **Frontend**: HTML, CSS, JavaScript

**2.2 Hardware Requirements**

* **Server**: Minimum 8GB RAM, 500GB SSD, 4-core processor
* **Client Devices**: Desktop or mobile devices with an internet connection

**Functional and Non-Functional Requirements for the Veterinary Hospital DBMS**

**1. Functional Requirements**

Functional requirements define the core operations and behavior of the system.

**Pet Registration**

* The system should allow users to register new pets with details such as name, species, breed, age, owner details, and registration date.
* The system should generate a unique Pet ID for each registered pet.
* Users should be able to update pet details, including medical history and vaccinations.

**Veterinary Doctors Management**

* The system should allow adding, updating, and removing veterinary doctor records.
* Each doctor should have a specialization field and be assigned to appointments.
* The system should allow searching for doctors based on their specialization.

**Nurses Management**

* The system should store nurse details such as name, department, and contact information.
* The system should allow assigning nurses to inpatient cases.
* The system should allow updating nurse details.

**Inpatient Management**

* The system should allow users to admit pets into the inpatient department.
* Users should be able to assign a doctor and a nurse to an inpatient case.
* The system should store admission and discharge dates, along with diagnoses.

**Outpatient Management**

* The system should allow scheduling outpatient visits for pets.
* Users should be able to assign a doctor to an outpatient visit.
* The system should record the reason for the visit and the treatment provided.

**Ward Management**

* The system should allow adding, updating, and removing wards.
* Each ward should have a unique ID, name, and capacity.
* The system should track ward occupancy to prevent overbooking.

**Pharmacy Management**

* The system should store medication details, including name and price.
* Users should be able to add, update, and remove medications from the pharmacy.
* The system should allow tracking medication stock levels.

**Billing Management**

* The system should generate bills for both inpatient and outpatient visits.
* The billing system should calculate charges based on medication and services provided.
* Users should be able to view, update, and print invoices for billing records.

**User Authentication and Access Control**

* The system should have different roles such as Admin, Doctor, Nurse, and Receptionist.
* Users should have role-based access to different system functionalities.
* Admins should be able to create and manage user accounts.

**2. Non-Functional Requirements**

Non-functional requirements define the quality attributes of the system.

**Performance Requirements**

* The system should be able to handle at least 100 concurrent users without performance degradation.
* Queries should return results within 2 seconds under normal load.

**Scalability**

* The system should support the addition of new features and an increasing number of users without major redesigns.

**Security Requirements**

* User authentication should be required to access system functionalities.
* Data should be encrypted, especially personal and medical information.
* The system should log all user activities for security and auditing.

**Usability**

* The system should have an easy-to-use interface with clear navigation.
* The system should be accessible via both desktop and mobile devices.

**Reliability**

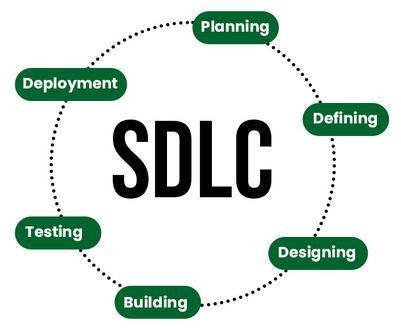
* The system should have a 99.9% uptime guarantee.
* The system should perform regular backups to prevent data loss.

**Maintainability**

* The system should be built using modular components to allow easy updates and maintenance.
* Database indexing should be used to optimize search and retrieval performance.

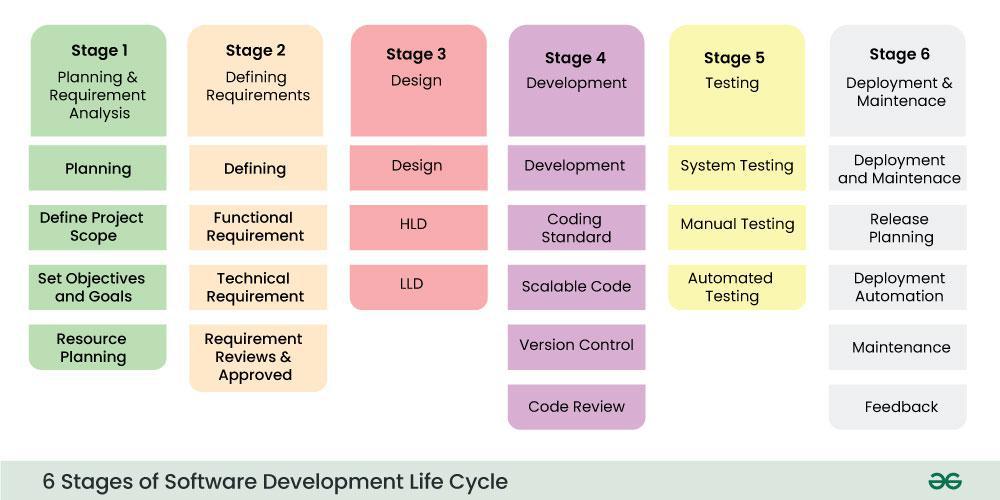
**Compliance**

* The system should comply with data protection regulations such as GDPR (if applicable).
* Medical records should be stored securely as per veterinary regulations.



**Stages of the Software Development Life Cycle**

SDLC specifies the task(s) to be performed at various stages by a software engineer or developer. It ensures that the end product is able to meet the customer’s expectations and fits within the overall budget. Hence, it’s vital for a software developer to have prior knowledge of this software development process. SDLC is a collection of these six stages, and the stages of SDLC are as follows:



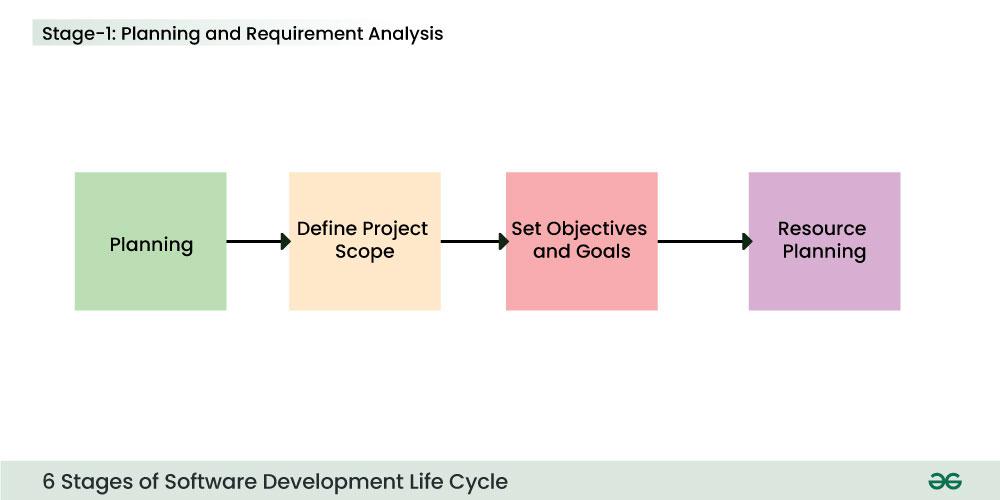
*Software Development Life Cycle Model SDLC Stages*

The [**SDLC Model**](https://www.geeksforgeeks.org/sdlc-models-types-phases-use)**involves six phases or stages** while developing any software.

**Stage-1: Planning and Requirement Analysis**

Planning is a crucial step in everything, just as in[software development](https://www.geeksforgeeks.org/software-development). In this same stage, [requirement analysis](https://www.geeksforgeeks.org/activities-involved-in-software-requirement-analysis)is also performed by the developers of the organization. This is attained from customer inputs, and sales department/market surveys.

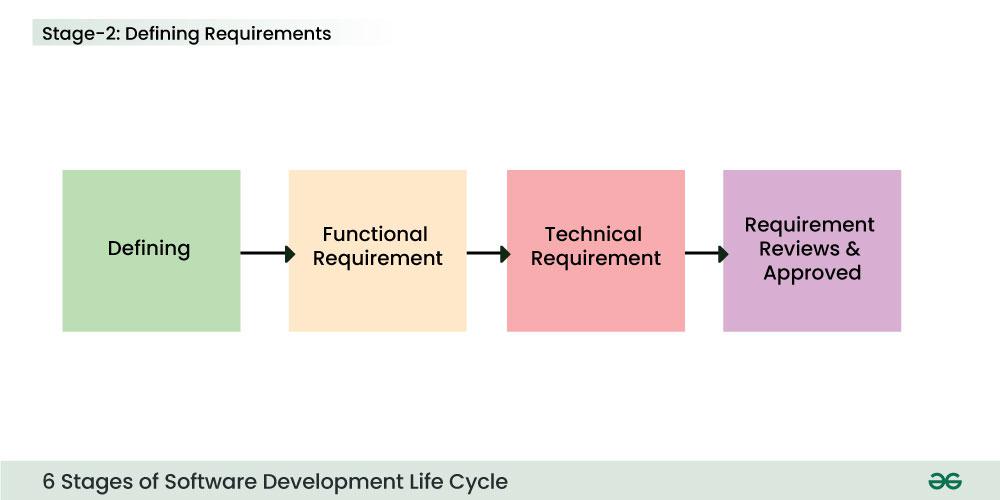
The information from this analysis forms the building blocks of a basic project. The quality of the project is a result of planning. Thus, in this stage, the basic project is designed with all the available information.



*Stage-1 : Planning and Requirement Analysis*

**Stage-2: Defining Requirements**

In this stage, all the requirements for the target software are specified. These requirements get approval from customers, market analysts, and stakeholders.   
This is fulfilled by utilizing SRS (Software Requirement Specification). This is a sort of document that specifies all those things that need to be defined and created during the entire project cycle.

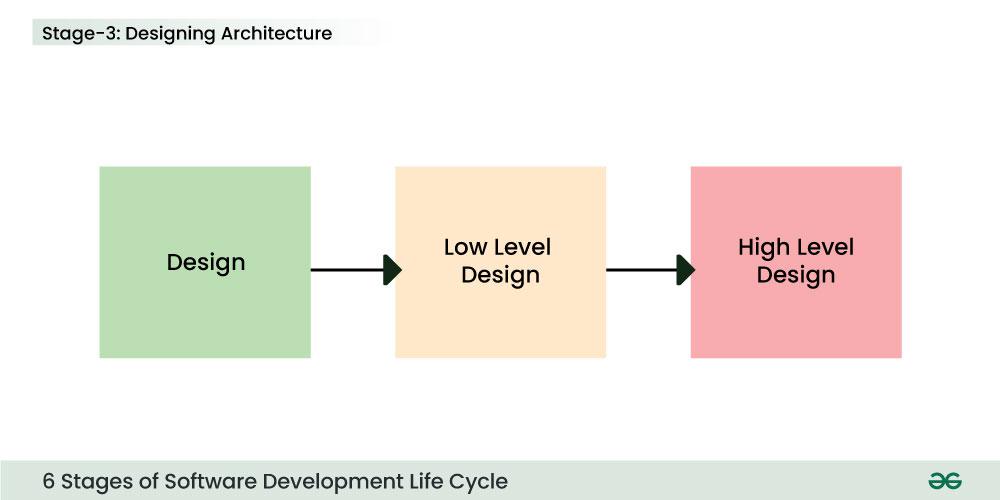


*Stage-2 : Defining Requirements*

**Stage-3: Designing Architecture**

[SRS](https://www.geeksforgeeks.org/software-requirement-specification-srs-format) is a reference for software designers to come up with the best architecture for the software. Hence, with the requirements defined in SRS, multiple designs for the product architecture are present in the Design Document Specification (DDS).

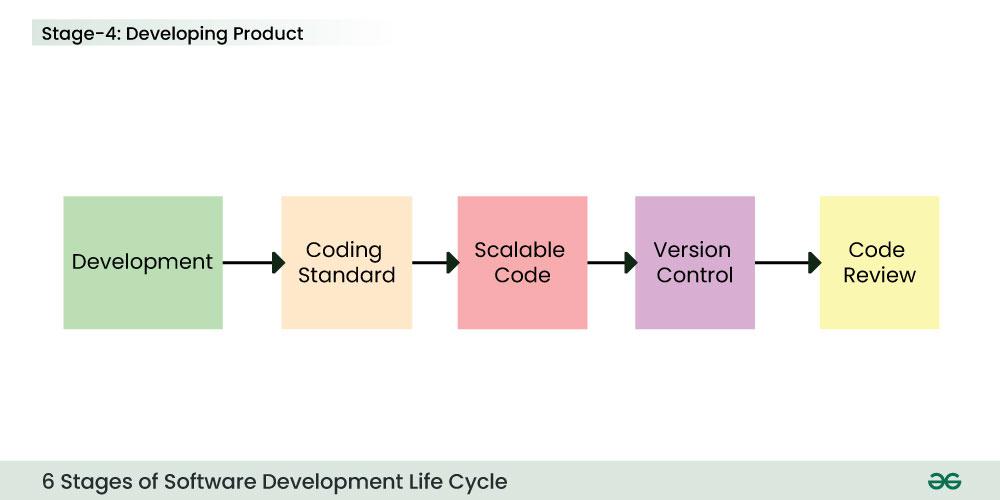
This DDS is assessed by market analysts and stakeholders. After evaluating all the possible factors, the most practical and logical design is chosen for development.



*Stage 3: Design*

**Stage-4: Developing Product**

At this stage, the fundamental development of the product starts. For this, developers use a specific programming code as per the design in the DDS. Hence, it is important for the coders to follow the protocols set by the association. Conventional programming tools like compilers, interpreters, debuggers, etc. are also put into use at this stage. Some popular languages like C/C++, Python, Java, etc. are put into use as per the software regulations.

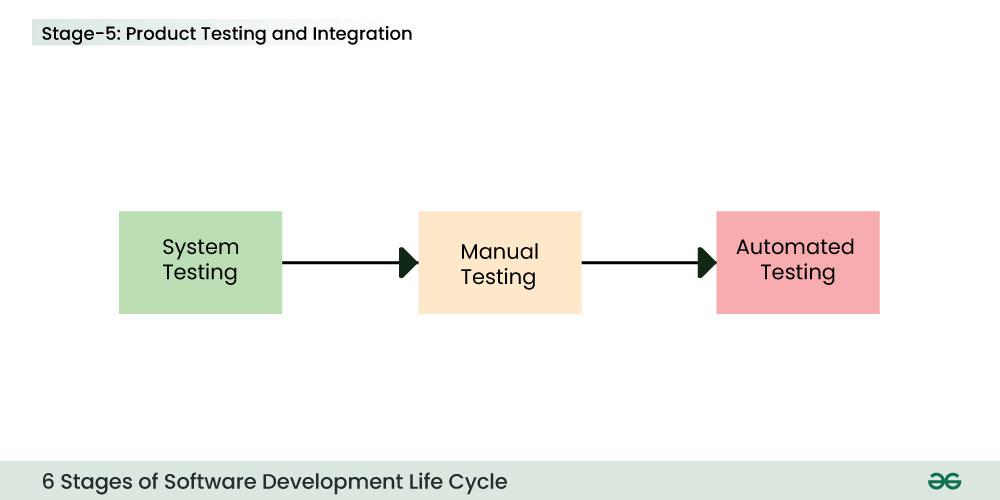


*Stage 4: Development*

**Stage-5: Product Testing and Integration**

After the development of the product, testing of the software is necessary to ensure its smooth execution. Although, minimal testing is conducted at every stage of SDLC. Therefore, at this stage, all the probable flaws are tracked, fixed, and retested. This ensures that the product confronts the quality requirements of SRS.

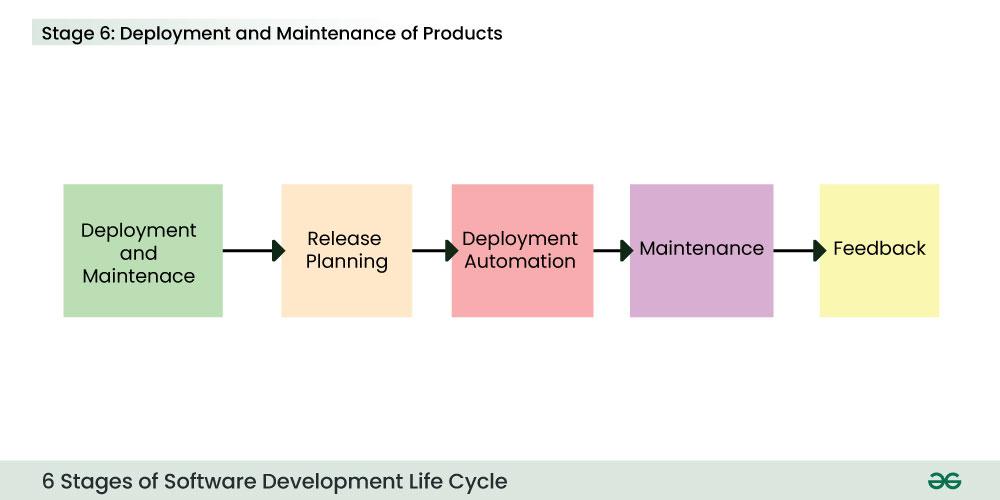
**Documentation, Training, and Support:**[Software documentation](https://www.geeksforgeeks.org/overview-software-documentation) is an essential part of the software development life cycle. A well-written document acts as a tool and means to information repository necessary to know about software processes, functions, and maintenance. Documentation also provides information about how to use the product. Training in an attempt to improve the current or future employee performance by increasing an employee’s ability to work through learning, usually by changing his attitude and developing his skills and understanding.



*Stage 5: Testing*

**Stage-6: Deployment and Maintenance of Products**

After detailed testing, the conclusive product is released in phases as per the organization’s strategy. Then it is tested in a real industrial environment. It is important to ensure its smooth performance. If it performs well, the organization sends out the product as a whole. After retrieving beneficial feedback, the company releases it as it is or with auxiliary improvements to make it further helpful for the customers. However, this alone is not enough. Therefore, along with the deployment, the [product’s supervision](https://www.geeksforgeeks.org/product-management).



*Stage 6: Deployment and Maintenance*